

AL/HR-TP-1996-0018



**USING A NEURAL NETWORK AND A STATISTICAL CLASSIFIER  
FOR AIRCRAFT FAULT DIAGNOSTICS**

**Shing P. Chu**

**HUMAN RESOURCES DIRECTORATE  
LOGISTICS RESEARCH DIVISION  
2698 G STREET  
WRIGHT-PATTERSON AFB, OHIO 45433-7022**

**AUGUST 1996**

**THIS QUALITY INSPECTED**

**19970103 014**

**ARMSTRONG  
LABORATORY**

**FINAL TECHNICAL PAPER FOR THE PERIOD MARCH 1995 TO AUGUST 1996**

**Approved for public release; distribution is unlimited**

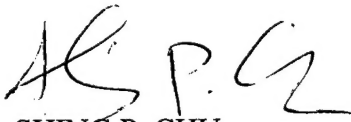
**AIR FORCE MATERIEL COMMAND  
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573**

## NOTICES

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation, or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.



SHING P. CHU  
Project Scientist



BERTRAM W. CREAM, Chief  
Logistics Research Division

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining, the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE August 1996	3. REPORT TYPE AND DATES COVERED Final Technical Paper - Mar 1995 to Aug 1996
4. TITLE AND SUBTITLE  Using A Neural Network and A Statistical Classifier for Aircraft Fault Diagnostics		5. FUNDING NUMBERS C - In-house PE - 62205F PR - 1710 TA - D2 WU - 01	
6. AUTHOR(S)  Shing P. Chu			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Armstrong Laboratory Human Resources Directorate Logistics Research Division 2698 G Street Wright-Patterson AFB OH 45433-7604		8. PERFORMING ORGANIZATION REPORT NUMBER  AL/HR-TP-1996-0018	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Armstrong Laboratory 2509 Kennedy Circle Brooks AFB TX 78235-5118		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES Armstrong Laboratory Project Scientist: Shing P. Chu, AL/HRGO, (513) 255-3871			
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited		12b. DISTRIBUTION CODE  -	
13. ABSTRACT (Maximum 200 words) In recent years, several techniques have been developed to create "intelligent" diagnostic aiding systems. Most of these systems, including the current Integrated Maintenance Information System (IMIS) diagnostics module, involve modeling the systems to be maintained. These systems have the disadvantage of requiring extensive efforts to develop them. A developing technology, neural networks, provides a promising alternative. Neural nets develop diagnostics strategies by learning from past experience with the system, and do not require extensive modeling. Neural networks are well suited to diagnostics applications. This paper presents: - A detailed description of a neural network based diagnostic system. - An explanation of a radial basis function(RBF) neural network architecture and its construction. - An explanation of the construction of a statistical classifier. - A description of data representation and method of system optimization. - Performance and experimental result.			
14. SUBJECT TERMS  artificial neural network      radial basis function aircraft diagnostics      leave-1-out bayesian classifier bench check serviceable		15. NUMBER OF PAGES  23	
		16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR

## CONTENTS

I.	INTRODUCTION .....	1
II.	BACKGROUND.....	1
III.	DIAGNOSTIC SYSTEM .....	2
IV.	NEURAL NETWORK ARCHITECTURE.....	3
V.	MAXIMUM LIKELIHOOD BAYESIAN CLASSIFIER.....	5
VI.	DATA REPRESENTATION .....	6
VII.	IMPLEMENTATION .....	8
VIII.	SYSTEM OPTIMIZATION .....	8
IX.	SYSTEM PERFORMANCE.....	9
X.	CONCLUDING REMARKS.....	10
	REFERENCES .....	10
	LIST OF ABBREVIATIONS.....	12
	APPENDIX A - LISTINGS	
1	Inputs and Outputs.. .....	13
2	Program to Generate Posterior Directly From Data.....	14
3	Program to Measure Performance Of The RBFN.....	15

## FIGURES

<u>Fig. No.</u>	<u>Page</u>
1	Diagnostics System Structure. .... 2
2	Radial Basis Function Network.. .... 4
3	An Example of TICARRS Data..... 6
4	Input and Output Representations..... 7
5	System Performance Optimization by Varying Sigma..... 9

## PREFACE

This report documents a study to examine the feasibility of using neural network technology for the development of troubleshooting procedures for on-board aircraft avionics systems. The research was conducted under Project 1710, Task D2, by the Logistics Research Division of the Armstrong Laboratory. Mr Shing Pak Chu was the principal investigator. Technical support was provided by the University of Dayton Research Institute and NCI, Inc.

## SUMMARY

In recent years, several techniques have been developed to create "intelligent" diagnostic aiding systems. Most of these systems, including the current Integrated Maintenance Information System (IMIS) diagnostics module, involve modeling the systems to be maintained. These systems have the disadvantage of requiring extensive efforts to develop them. A developing technology, "neural networks," provides a promising alternative. Neural nets develop diagnostics strategies by learning from past experience with the system, and do not require extensive modeling. Neural networks are well suited to diagnostics applications.

A Radial Basis Function Network (RBFN) was built using Matlab Neural Networks Toolbox. A diagnostic system was modeled from the F-16 Fire Control Radar (FCR) data. The neural network was trained using FCR maintenance records obtained from the F-15 System Program Office. Evaluation of the neural nets indicated that, although the neural nets were able to successfully isolate faults in the testbed system, the demonstrated accuracy for fault diagnosis did meet the objectives of the test.

## I. INTRODUCTION

The traditional diagnostic methods for aircraft maintenance using technical manuals are costly to author and often fail to isolate the cause of the system failure; thus, these traditional methods impact mission readiness and increase maintenance costs. High field maintenance hours and false removals are often caused by incorrect diagnosis. The accuracy of fault diagnosis could be improved by the use of maintenance historical data. However, this data is difficult to access and is seldom used. A diagnostic system is needed that is capable of learning from historical data and using this information to help in identifying the faulty unit and correctly predicting the cause of false removals, such as Line Replaceable Units (LRU) that Bench Check Serviceable (BCS) when tested in the shop depot. Artificial neural networks provide a possible solution to this fault diagnosis problem. This paper describes the development of a diagnostic system using an artificial neural network combined with a Bayesian maximum likelihood classifier to provide the required diagnostic capability without requiring the services of costly maintenance experts. The background describes the problem and briefly introduces the artificial neural network approach. For readers without a lot of knowledge of artificial neural networks, a list of references is provided. Section III contains a description of the diagnostic system and its operation. Section IV concerns one of the components, the radial basis function network (RBFN), including its architecture and mathematical relevancy. The other main component, the maximum likelihood classifier, is featured in Section V. Section VI contains a description of the data representation scheme. The "Implementation" section, Section VII, includes a description of system inputs, outputs, and training and testing methods, while Section VIII concerns system optimization. Finally, system performance was calculated in terms of accuracy and execution time in Section IX.

## II. BACKGROUND

BCS occurs when the reported faulty component checks good when tested in the back shop. BCS occurs at the maintenance shop level and is always the result of unnecessary removal of LRU. Early knowledge that an LRU has a history of frequent BCSs and is likely to BCS can help the maintenance technician to modify his diagnostic procedures, thus reducing unnecessary removal of LRU's. A well-trained technician can tell from the maintenance history of the LRU whether it is likely to BCS, thus providing this project with a set of sample data. This report presents a process for developing a practical fault classifier that uses neural network technology and historical data to identify faults in today's military aircraft systems. The F-16 FCR data was used as the testbed for this study.

The term "artificial neural networks" refers to any computing architecture that consists of massively parallel, interconnected, simple "neural" processors. It was postulated that a model with a structure similar to a biological neural network could have similar intelligence functions. By carefully emulating the brain, artificial neural networks have exhibited such brain-like characteristics as the ability to learn from experiences,



generalize on knowledge, and perform information extraction. Neural networks can be used as computational models based on linear system theories and design methodologies.

### III. DIAGNOSTIC SYSTEM

The diagnostic system is designed to identify the faulty component and the type (normal, lemon, or bad actor) of fault that is occurring. The structure of the diagnostic system is depicted in Figure 1. Maintenance Fault Lists (MFL) are entered into a neural network that has been trained to recognize the difference between symptom patterns and identify the faulty LRU. Once the LRU is identified, the same symptom is entered into a lower level Bayesian classifier to classify the identified LRU as a "normal," "lemon," or "bad actor." Three additional pieces of information are required. The posterior matrices are generated from the product of "a prior" and conditional probability matrices. From Baye's rule, we form the classifier on the basis of "opportunity loss" x "posterior," and a "payoff" matrix is formed. The payoff matrix contains a decision that takes into account the likelihood of occurrence and the opportunity loss or misclassification associated with a particular set of symptoms or MFL codes. We interpret the payoff using the identified LRU, as an indicator, and Baye's rule declares the category winner (i.e., normal, lemon, or bad actor) that has the highest payoff.

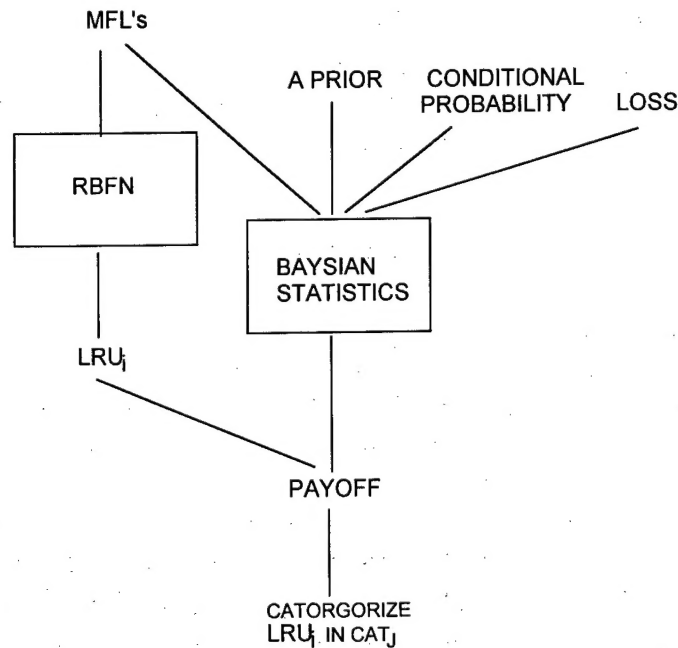


Figure 1. Diagnostic System Structure

#### IV. NEURAL NETWORK ARCHITECTURE

A standard radial basis function (RBF) network is used to perform the pattern classification task. The goal is to identify the offending LRU. Cover's theorem on the separability of patterns states that a complex classification problem cast in high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space (Cover, 1965). Taking Cover's stance that learning is a matter of reconstructing the hypersurface in a multidimensional space given available training data, generalization is the use of this surface to interpolate the test data. Given that input data is high-dimensional and with unknown correlation, the use of an RBF network is justified.

Consider the Gaussian function,

$$G(\|x - t_i\|) = e^{-\frac{\|x - t_i\|^2}{2\sigma^2}}, \quad i = 1, 2, \dots, k$$

where  $x$  is input and  $t_i$  is the  $i$  measurement centered at the data point.

Assume the desired outputs of the described problem have a nonzero mean. The standard deviation,  $\sigma$ , is used to fine tune the width of the Gaussian curves; therefore, the input-output relation of the RBFN is defined by: (Haykin, 94)

$$y(x) = \sum_{i=1}^k w_i G(\|x - t_i\|) + b$$

where  $w$  is the weight matrix.

To have a perfect fit of the training data,

$$y(x_j) = d_j, \quad j = 1, 2, \dots, p$$

where  $x_j$  is an input vector and  $d_j$  is the desired output.

For convenience, the equations are put in matrix form:

$$g_{ji} = G(\|x_j - t_i\|), \quad j = 1, 2, \dots, p; i = 1, 2, \dots, k.$$

Therefore,

$$Gw = d$$

where

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1p} \\ g_{21} & g_{22} & \cdots & g_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ g_{k1} & \cdots & \cdots & g_{kp} \end{bmatrix}.$$

All parameters of this network are known except the weight matrix  $w$ , and there are more training samples than inputs, that is,  $G$  is not squared,  $k > p$ . We can solve  $w$  by solving an overdetermined linear system of the form,

$$w = G^+ d,$$

where  $G^+$  denotes the pseudoinverse of  $G$ .

By casting an orthogonal projection of  $d$  to a hyperplane that is colinear in the direction of  $w$ , we form  $Gw$ . Any  $w$  that minimizes the residue,  $r = d - Gw$ , is a least squares solution to this overdetermined system. The desired equation is  $d - Gw = 0$ , or  $Gw = \tilde{d}$ . Multiply both sides by  $G^T$ , and  $G^T Gw = G^T d$ , and finally  $w$  can be expressed as

$$w = (G^T G)^{-1} G^T d.$$

This form is computationally efficient because matrix  $G^T G$  is symmetric positive definite (SPD).

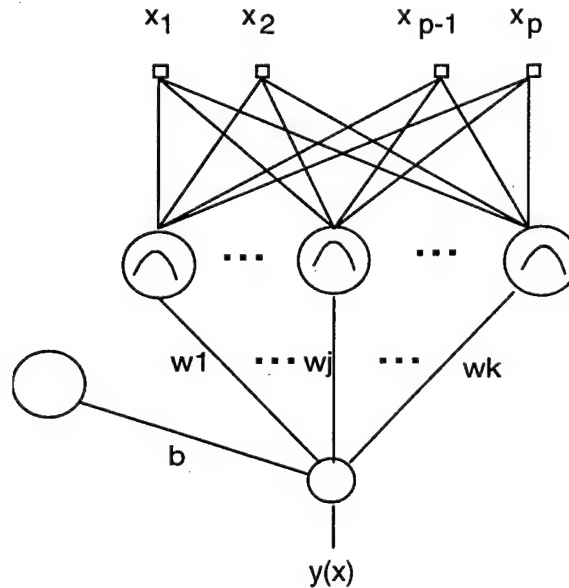


Figure 2. Radial Basis Function Network

The RBF network in Figure 2 has two layers: a hidden layer and an output layer. The hidden layer calculates the Euclidean norm,  $\|x_j - t_i\|$ . This layer calculates the distance

between input vector to each sample data vector. Each neuron will output a value according to how close the input vector is to each neuron's sample data vector. The results are passed to an RBF, in this case, an exponential function. This forms the outputs of the hidden neurons. The output layer contains a weight matrix,  $w$  and  $b$ , the bias. The weights,  $w$ , can be solved by  $w = G^+d$  as described above. Note that  $b$  is inserted as part of  $w$ , (i.e.,  $w = [w_1 \ w_2 \ \dots \ w_k \ b]$ ).

## V. MAXIMUM LIKELIHOOD BAYESIAN CLASSIFIER

A maximum likelihood classifier is constructed to classify the offending LRU to be either a "normal," "lemon," or "bad actor." "Normal" simply means the LRUs only require normal maintenance procedures (e.g., repair, removal and replacement, and so forth). A "lemon" defines the LRU as having multiple incidents of BCSs occurring in different aircraft. A "bad actor" also signifies multiple BCSs, but only occurring erratically in some aircraft. A full description is included in a report of the Bad Actor Program from the Air Force F-15 System Program Office (DRC, 94). To build the classifier, we need "a priori," "conditional probability," and "opportunity loss" matrices (McClave, 85). Their relationships are as follows.

Assuming there are  $m$  LRUs,  $n$  categories and  $q$  MFLs, the following definitions apply.

$P_{rij}$  - A priori, probability of LRU<sub>*i*</sub> in category<sub>*j*</sub>, this matrix is sized  $m \times n$ .

$P_{cij}$  - Probability of MFL<sub>*k*</sub> conditioned on LRU<sub>*i*</sub> belongs to category<sub>*j*</sub>, this matrix is sized  $m \times n \times q$ .

$P_{pij}$  - Posterior, probability of LRU<sub>*i*</sub> in category<sub>*j*</sub> conditioned on MFL<sub>*k*</sub>, this matrix is sized  $m \times n \times q$ .

From Baye's rule, for each MFL<sub>*k*</sub>,

$$P_{pij} = P_{rij} P_{cij}$$

For each misclassification, there is always some associated cost penalizing the mistake. The cost can be in terms of dollar amounts, time wasted, or a combination of these factors:

$$l_{ij} = \alpha C_{ij} + \beta T_{ij}$$

where  $l_{ij}$ ,  $C_{ij}$ , and  $T_{ij}$  are the opportunity loss, cost, and time resulting from misclassification of LRU<sub>*i*</sub> in fault category<sub>*j*</sub>; respectively, and  $\alpha$  and  $\beta$  are weighting parameters to balance the cost and time criteria. In one extreme case where the maintenance time is the dominant criterion (usually in war time),  $\alpha = 0$ . In the other extreme case where the maintenance cost is the dominant criterion (usually in peace

time),  $\beta = 0$ . The cost coefficients form an  $m \times n$  cost coefficient matrix. Likewise, the time coefficients form an  $m \times n$  time coefficient matrix. The sum of the two form the opportunity loss matrix. The opportunity loss matrix is also conditioned on  $MFL_k$  and will be discussed in the data section. Finally, the "payoff" matrix is defined as

$$P_{payoff} = P_{pij} l_{ij}$$

Since both the posterior and opportunity loss matrices are conditioned on  $MFL_k$ ,  $MFL_k$  is used as an indicator to summon  $P_{payoff}$  on demand.

Interpretation of  $P_{payoff}$  is straightforward.  $P_{payoff}$  contains the posterior probabilities scaled by the opportunity losses. Baye's rule dictates that the largest posterior shall win; therefore, the identified  $LRU_i$  is used as an indicator to locate a row in the payoff matrix and find the highest value in that row. The column will suggest the winning category.

## VI. DATA REPRESENTATION

Data was acquired through Tactical Interim Core Automated Maintenance System (CAMS) And Reliability and Maintainability Information System (REMIS) Reporting System (TICARRS) of the F-15 SPO. This data was obtained for the F-16 C/D block 40 FCR from the period of August 1993 to August 1994 covering these bases: Kirkland, Eielson, Hill, Luke, Moody, and Shaw. Three sets of data were collected and analyzed. The first set was collected by querying all records with a string containing "MFL" and fault reporting code (FRC). This set contains 466 records, including Work Unit Code, Job Control Number, Year, Day, Action Taken, Howmal Code, Discrepancy, and Corrective Action (see Fig. 3.). This set of data was used to train the RBFN to identify the offending LRU. Once the RBFN was trained, the data was discarded.

W U C	J C N	Y R	DAY	A T	HOWMAL	FRC	
74APD	0104006	94	10	R	103	77	
						DISCREPANCY :	AIR ABORT FOR O
							FOR WOULD PWR DI
							TIMES BUT COM
							CYCLING SWITCH
						CORRECTIVE ACTION :	T/S SYS, FOUND (
							AND DID HAVECU
							FCR BIT CK AND

Figure 3. An Example of TICARRS Data

Only MFLs in the discrepancies and the repaired LRUs were used for the classification. We gave a binary coding (Apolloni, 1990) to the set of all possible inputs (MFLs) and all possible outputs (LRUs), (see Figure 4). Records judged not relevant for fire control radar were excluded and duplications of records were eliminated. There are 137 inputs and 7 outputs (see Listing 1.). Each possible MFL represents an index

position in the input vector and each possible LRU represents an index position in the output vector. The output is coded "one-of-N," which means that only one LRU is trained at a time.

## Data Elements : MFL and LRU

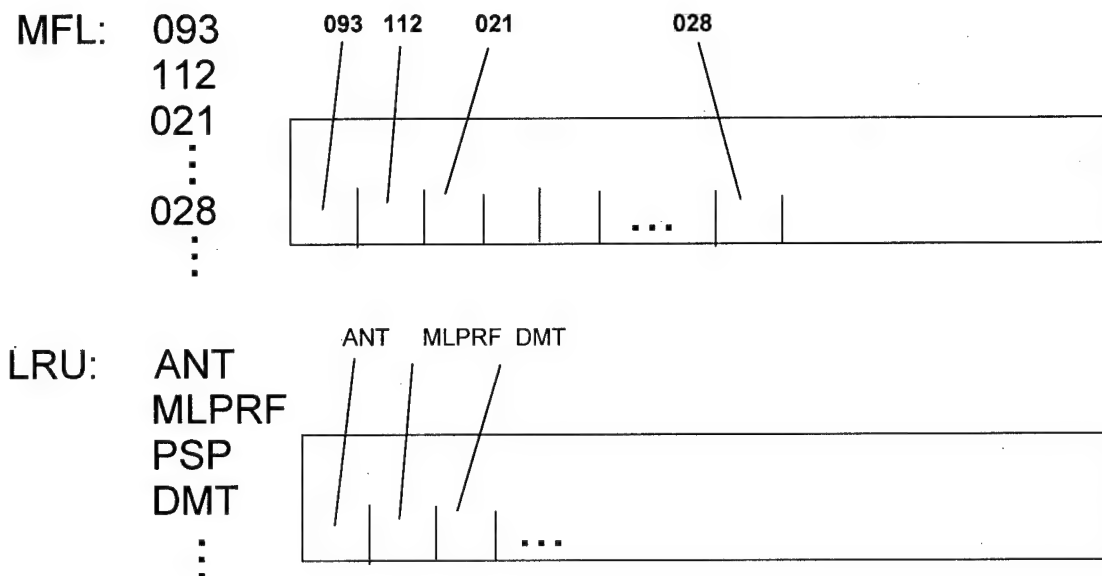


Figure 4. Input and Output Representations

When attempting to solve the BCS problem, we found that the first set of data was inadequate because BCS always implies multiple LRU removals and the first set of data did not contain information on LRUs installed or removed. We collected a second set of data which contained the required information for the period March 1995 to August 1995. Only 56 records were collected because TICARRS maintains records of LRU part numbers for only six months. The same types of information were collected, except we added information concerning aircraft tail numbers and part numbers of LRU installed or removed. Based on experience, a maintenance expert analyzed this data and assigned either a "normal," "bad actor," or "lemon" category to each LRU encountered.

A third set of data was collected to construct the "opportunity loss matrix", resulting from misclassification of  $LRU_i$  in fault category  $j$ . The opportunity loss can be time, cost, or subjectively determined. The actual cost of the misclassifications was used in this matrix.

## VII. IMPLEMENTATION

The RBFN was implemented using MATLAB neural networks toolbox on a 100 MHz pentium PC. The input layer has 137 neurons, the hidden layer has 465 neurons (see the optimization section below), and the output layer has 7 neurons. The first set of data was reformatted in a text file, "mldata.txt," from which the training input,  $P$ , and the desired output,  $T$ , matrices were extracted. We trained the network by presenting the matrix  $P$  and desired output matrix  $T$  to the network. Setting the 'weights' of each hidden neuron to each sample input, the network calculates the Euclidean norm,  $\|x - t_i\|$ , and then  $g_{ij}$  is calculated. The desired output,  $d$ ,  $w = (G^T G)^{-1} G^T d$  was computed and the RBFN was trained. By presenting a binary vector with 137 elements to the input, the network will estimate a vector of size  $1 \times 7$  by computing a new  $Gw$ . This output is the sum of probability density functions (PDF), and can be used for ranking purposes.  $w$  is precomputed during training.

The "a priori" matrix and "conditional probability" matrices were extracted from the second set of data.  $P_{rij}$  was formed by counting the frequency of occurrence of  $LRU_i$  belonging to category  $j$ . A  $7 \times 3$  matrix was formed and normalized by dividing each row by its highest value.  $P_{cijk}$  was formed by counting the frequency of occurrence of  $MFL_k$  when  $LRU_i$  belongs to category  $j$ . A  $7 \times 3 \times 137$ , normalized, three dimensional (3-D) array was used to store  $P_{cijk}$ .  $P_{pijk}$  was handled slightly different than above, although  $P_{pijk}$  is also conditioned on  $MFL_k$ . Given an  $MFL_k$ ,  $P_{pijk}$  can be calculated from  $P_{rij} P_{cij}$  on demand; therefore, a 3-D array storage is unnecessary.  $P_{pijk}$  is sized  $7 \times 3$ . An alternative is to obtain  $P_{pijk}$  directly from the data file, (see Listing 2). The 'opportunity loss' matrices were formed using a 3-D array.  $P_{rij}$ ,  $P_{cijk}$ , and  $l_{ijk}$  form  $P_{payoff}$  for any given  $MFL_k$ . As multiple MFLs occurred, all  $P_{payoff}$ 's were added. Finally,  $P_{payoff}$  is normalized by  $P_{payoff} / \max(P_{payoff})$ .

## VIII. SYSTEM OPTIMIZATION

Recall that the standard deviation,  $\sigma$ , is used to optimize the RBFN by adjusting the width of the Gaussian curves. A technique developed by the Ford Motor Company called "leave-1-out" (Marko, 1990) was used to estimate the optimized  $\sigma$ . It is presumed that the training data is representative of the data that span the multi-dimensional space sufficiently so that the classifier can generalize usefully. In this context, generalization means that the output produced by RBFN was not among those that it was trained from. The "leave-1-out" method is described as follows:

The entire set of data samples except one was used to train the RBFN using a  $\sigma$ . After the network was trained, the network was tested by using it to identify the fault for the data sample not used for training. The output was recorded. This process was repeated 466 times, and a different sample was held out each time. The percentage of the total correct solution was recorded. The process was completed for one  $\sigma$ . To optimize

the system, a range of  $\sigma$ 's were used. The optimal  $\sigma$  was found to be  $0.8326/sc$ , where  $sc = 1.5$  (see Figure 5 and Listing 3).

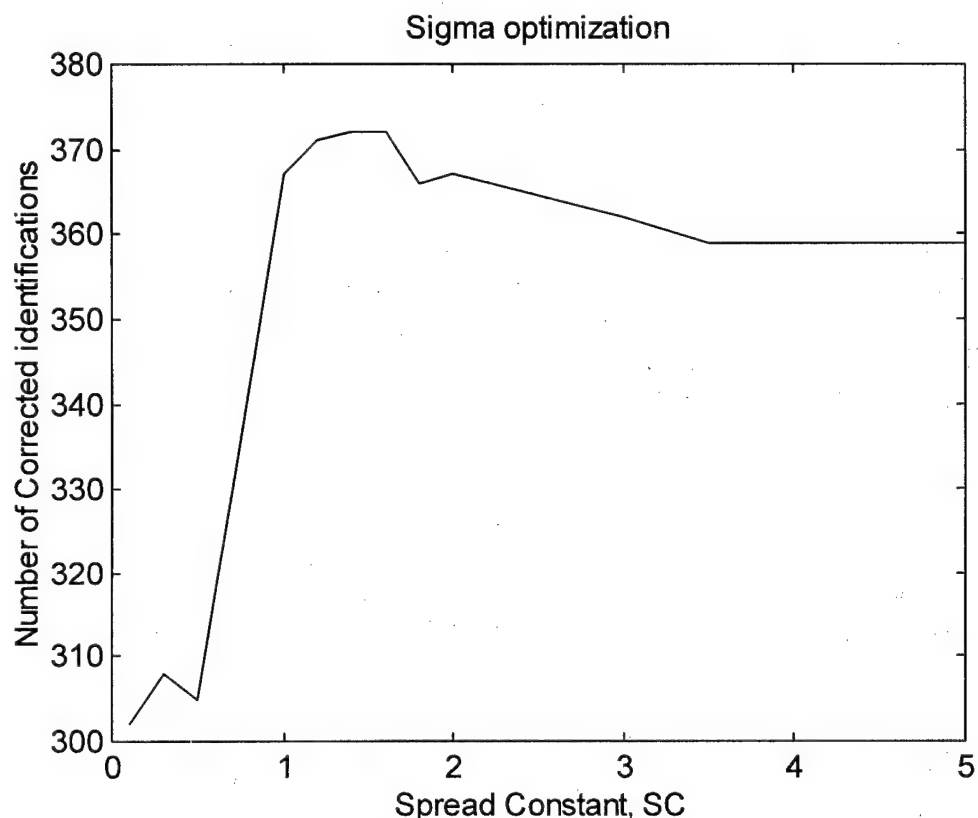


Figure 5. System Performance Optimization by Varying Sigma

## IX. SYSTEM PERFORMANCE

While using the "leave-1-out" method of optimization, the accuracy of the RBFN can be measured simultaneously. With the optimal  $\sigma$ , the RBFN produced 372 correct answers, which included guessing the top ranked LRU that was actually bad with an 80% accuracy rate. The RBFN ranked the faulty LRU first or second with an accuracy of 86.48%, or 403 correct answers. The total training time for one  $\sigma$  is  $6.5634 \times 10^4$  seconds, or 18.23 hours. For a single input vector, the network execution time is 0.55 seconds. As for the BCS classifications, because of the little amount of data gathered, a stochastic approach was not feasible. A statistical approach was used.  $P_{rij}$ ,  $P_{cijk}$  and  $I_{ijk}$  function as lookup tables to generate  $P_{payoff}$  dynamically. Since all operations of the matrices are done element-by-element, the computations are very efficient.



## X. CONCLUDING REMARKS

A deterministic Bayesian classifier supplemented by a RBFN was developed for aircraft fault classification (normal, lemon, or bad actor) for concluding if an LRU is indeed defective. The RBFN is a parallel computational model designed to identify the defective LRU. It was disappointing that the RBFN only provided an 80% accuracy rate. It is hoped that the RBFN can operate at an accuracy rate beyond 90% to be practical. Theoretically, as more data accumulates, the system accuracy will improve. As the network grows in size, network execution speed will suffer. A version of RBFN with a clustered hidden layer can be developed to counter the size problem. Another setback is the extensive data analysis time. Future research must include the automatic generation of training data from raw data. Nevertheless, the neural net approach minimizes the system modeling effort and eliminates writing expert rules. The Bayesian classifier offers a solution to the BCS problems based on statistical measures.

## REFERENCES

Apolloni, B., Avanzini, G., Cesa-Bianchi, N. and Ronchini, G., 1990. *Diagnosis of Epilepsy via Backpropagation*. Proceedings of the IJCNN-90-Wash. D.C., Washington, D.C.

Chen, S., Cowan, C. F. N. and Grant, P. M., March 1991. *Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks*. IEEE Transactions on neural networks, Vol. 2, No. 2.

Cover T. M., 1965. *Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition*. IEEE Transactions on Electronic Computers EC-14, pp. 326-334.

Dynamics Research Corporation. Feb. 1994. *F-15 Bad Actor Program*. ASC/VFLS, Wright-Patterson AFB, OH.

Haykin, Simon, 1994. *Neural Networks. A Comprehensive Foundation*. Macmillan College Publishing Company.

Marko, K. A., Feldkamp, A. and Puskorius, G. V., 1990. *Automotive Diagnostics Using Trainable Classifiers: Statistical Testing and Paradigm Selection*. Proceedings of the IJCNN-90-San Diego, San Diego, Calif.

McClave, J. T., Benson, P. G., 1985. *Statistics for Business and Economics*. Collier Macmillan Publishers, third edition.

McDuff R. J., Simpson P. K., and Gunning D., 1989. *An investigation of neural networks for F-16 fault diagnosis: system description*. Proceedings of IEEE AUTOTEST Conference, pp. 351-357.

## List Of Abbreviations

AL/HRG	Armstrong Laboratory/ Logistics Research Division
BCS	Bench Check Serviceable
CAMS	Core Automated Maintenance System
DRC	Dynamics Research Corporation
FCR	Fire Control Radar
FRC	Fault Reporting Code
IMIS	Integrated Maintenance Information System
LRU	Line Replaceable Unit
MFL	Maintenance Fault List
PDF	Probability Density Function
RBF	Radial Basis Function
RBFN	Radial Basis Function Network
REMIS	Reliability and Maintainability Information System
SC	Spread Constant
SPD	Symmetric Possible Definite
SPO	System Program Office
TICARRS	Tactical Interim CAMS And REMIS Reporting System

# APPENDIX A LISTINGS

## LISTING 1: Inputs and Outputs

MFL	001	MFL	059	MFL	117	MFL	245
MFL	002	MFL	060	MFL	119	MFL	246
MFL	003	MFL	061	MFL	120	MFL	248
MFL	004	MFL	062	MFL	122	MFL	249
MFL	005	MFL	064	MFL	124	MFL	252
MFL	006	MFL	065	MFL	125	MFL	253
MFL	007	MFL	068	MFL	126	MFL	255
MFL	008	MFL	069	MFL	127	MFL	269
MFL	009	MFL	070	MFL	128	MFL	270
MFL	010	MFL	071	MFL	184	MFL	271
MFL	020	MFL	074	MFL	200	MFL	272
MFL	021	MFL	084	MFL	203	MFL	273
MFL	022	MFL	085	MFL	205	MFL	274
MFL	024	MFL	087	MFL	209	MFL	276
MFL	026	MFL	088	MFL	212	MFL	280
MFL	028	MFL	089	MFL	213	MFL	284
MFL	029	MFL	090	MFL	215	MFL	285
MFL	035	MFL	091	MFL	216	MFL	286
MFL	036	MFL	092	MFL	218	MFL	288
MFL	037	MFL	093	MFL	219	MFL	290
MFL	038	MFL	094	MFL	220	MFL	293
MFL	039	MFL	095	MFL	222	MFL	294
MFL	041	MFL	096	MFL	223	MFL	295
MFL	042	MFL	097	MFL	224	MFL	298
MFL	043	MFL	098	MFL	226	MFL	302
MFL	044	MFL	101	MFL	231	MFL	308
MFL	046	MFL	102	MFL	235	MFL	310
MFL	052	MFL	103	MFL	236	MFL	316
MFL	053	MFL	104	MFL	237	MFL	318
MFL	054	MFL	107	MFL	238	MFL	324
MFL	055	MFL	108	MFL	240	MFL	327
MFL	056	MFL	112	MFL	242	MFL	331
MFL	057	MFL	114	MFL	243	MFL	335
MFL	058	MFL	116	MFL	244	MFL	338
						MFL	341

LRU	1	ANTENNA
LRU	2	MODULAR LOW POWER RADIO FREQ UNIT
LRU	3	DUAL MOD. TRANSMITTER
LRU	4	PROGRAMMABLE SIGNAL PROCESSOR
LRU	5	WAVEGUIDE ASSY.
LRU	6	ABS. PRESS RELIEF VALUE
LRU	7	WIRING

## LISTING 2: Program to Generate Posterior Directly From Data

```
% This illustrates how to generate posterior for all MFL's
% k=1..137 from a text file.
% Variable s stores the posterior for the kth MFL index.
% Variable ns stores the normalized posterior*for the kth MFL.
% The variable acc accumulates all posterior's.
% To use this routine, first adjusts values of m, l, a and s
% to the data file size, then take out the k-loop,
% and substitute k for the MFL index. At last, take out the
% accumulator and the diary.
%
% Creator: Shing P.Chu, The Air Force Armstrong Laboratory
% Date: Aug 24, 95
```

```
load 'mldata.dat'
m = mldata(:,1:137);
l = mldata(:,138:144);
a = mldata(:,145);
s = zeros(7,3);
acc = s;

for k = 1:137
    for i = 1:7
        r = m(:,k) + l(:,i);
        rr=find(r==2);
        r = zeros(size(r));
        r(rr) = ones(size(rr));
        p = r;
        az = p .* a;
        rrr = r + az;
        s(i,1) = length(find(rrr==2));
        s(i,2) = length(find(rrr==3));
        s(i,3) = length(find(rrr==4));
        if sum(s(i,:)) == 0
            ns(i,:) = zeros(size(s(i,:)));
        else ns(i,:) = s(i,:)/sum(s(i,:));
        end;
    end;
    acc = acc + s;
    diary on;
    k
    [s ns]
    diary off;
end;
diary on;
acc
diary off;
```

### LISTING 3: Program to Measure Performance Of The RBFN

```
% This program implements the hold-1-out method of training and evaluation
% of the exact form of RBFN. There are 137 inputs, 465 hidden neurons and
% 7 outputs. There are a total of 466 records for both training and
% testing.
%
% Creator: Shing P.Chu, The Air Force Armstrong Laboratory
% Date: Aug 30, 95

%   DEFINING VECTORS TO BE CLASSIFIED
%   =====

load 'mldata.txt';
r = randperm(466);          % randomly scrambles the data records, may
                           % be unnecessary for RBFN.
rdata = mldata(r(:,:),:);  % rdata becomes the new data set.

%for sigma = 1.2:0.3:1.8
sigma = 1.5;                % sigma optimal at 1.5.
mmldata = rdata';           % use mmldata from now on.
t0 = clock;                 % start the time count.
corr = 0;
for i = 1 : 466              %
learn = mmldata(:, 2:466);    %
test = mmldata(1:137, 1);    %           hold-1-out
desire = mmldata(138:144, 1); %
mmldata = [learn mmldata(:,1)]; %

P = learn(1:137, :);         % training inputs

%   Here are the classes these vectors fall into.
T = learn(138:144, :);       % target outputs

% INITIALIZE NETWORK ARCHITECTURE
%=====
% sc = [10 100 0.02 1];
% [w1,b1,w2,b2,nr,tr] = solverb(P,T);
% [w1,b1,w2,b2] = solverbe(P,T,sigma);

%   Here we present the input vector

a = simurb(test,w1,b1,w2,b2);

diary
sigma
i
a = full(a);                % begin evaluate the hold-1-out
rr=find(a==max(a));         %
a = zeros(size(a));         %
a(rr) = ones(size(rr));     %
if a == desire              %
    corr = corr + 1;        % accumulates the correct answers
end;                        %
corr                       %
[a desire]                  %
end;                        %
etime(clock,t0)             % record the time used
diary off
%end;
```